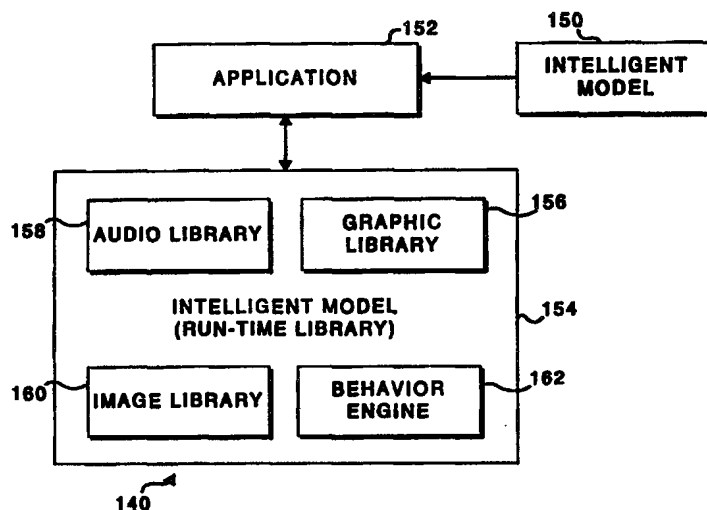




## INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

<b>(51) International Patent Classification <sup>6</sup> :</b>  <b>G06T 17/00</b>	<b>A1</b>	<b>(11) International Publication Number:</b> <b>WO 99/05651</b>  <b>(43) International Publication Date:</b> 4 February 1999 (04.02.99)
<b>(21) International Application Number:</b> PCT/CA98/00696  <b>(22) International Filing Date:</b> 16 July 1998 (16.07.98)  <b>(30) Priority Data:</b> 08/900,488                      25 July 1997 (25.07.97)                      US  <b>(71) Applicant (for all designated States except US):</b> SOFTIMAGE INC. [CA/CA]; Suite 500, 3510 Boulevard St-Laurent, Montreal, Quebec H2X 2V2 (CA).  <b>(72) Inventor; and</b> <b>(75) Inventor/Applicant (for US only):</b> CAJOLET, Claude [CA/CA]; 3857 Drolet, Montreal, Quebec H2W 2L3 (CA).  <b>(74) Agents:</b> STRATTON, Robert, P. et al.; Gowling, Strathy & Henderson, Suite 4900, Commerce Court West, Toronto, Ontario M5L 1J3 (CA).		<b>(81) Designated States:</b> AL, AM, AT, AU, AZ, BA, BB, BG, BR, BY, CA, CH, CN, CU, CZ, DE, DK, EE, ES, FI, GB, GE, GH, GM, HR, HU, ID, IL, IS, JP, KE, KG, KP, KR, KZ, LC, LK, LR, LS, LT, LU, LV, MD, MG, MK, MN, MW, MX, NO, NZ, PL, PT, RO, RU, SD, SE, SG, SI, SK, SL, TJ, TM, TR, TT, UA, UG, US, UZ, VN, YU, ZW, ARIPO patent (GH, GM, KE, LS, MW, SD, SZ, UG, ZW), Eurasian patent (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European patent (AT, BE, CH, CY, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE), OAPI patent (BF, BJ, CF, CG, CI, CM, GA, GN, GW, ML, MR, NE, SN, TD, TG).  <b>Published</b> <i>With international search report.</i> <i>Before the expiration of the time limit for amending the claims and to be republished in the event of the receipt of amendments.</i>

**(54) Title:** INTELLIGENT MODEL FOR 3D COMPUTER GRAPHICS**(57) Abstract**

Three-dimensional (3D) intelligent models are provided with data and programming instructions that enable the intelligent models to implement a predefined behavior in response to an event. The intelligent models include the conventional data that define the shape and appearance of the object, but in addition can access or include parameters to exhibit behavior corresponding to that of the object represented by the intelligent model. For example, an intelligent model of a television includes controls that respond to changes causing the image on a screen of the intelligent model or the sound produced by its speaker to change in an animation just as would be expected of a television. A library of intelligent models can be provided with plural sets of behaviors applicable to a class of intelligent models. Thus, an intelligent model representing a human character may be provided with a set of behaviors for an athlete so that the intelligent model responds to events in an athletic manner.

***FOR THE PURPOSES OF INFORMATION ONLY***

Codes used to identify States party to the PCT on the front pages of pamphlets publishing international applications under the PCT.

<b>AL</b>	Albania	<b>ES</b>	Spain	<b>LS</b>	Lesotho	<b>SI</b>	Slovenia
<b>AM</b>	Armenia	<b>FI</b>	Finland	<b>LT</b>	Lithuania	<b>SK</b>	Slovakia
<b>AT</b>	Austria	<b>FR</b>	France	<b>LU</b>	Luxembourg	<b>SN</b>	Senegal
<b>AU</b>	Australia	<b>GA</b>	Gabon	<b>LV</b>	Latvia	<b>SZ</b>	Swaziland
<b>AZ</b>	Azerbaijan	<b>GB</b>	United Kingdom	<b>MC</b>	Monaco	<b>TD</b>	Chad
<b>BA</b>	Bosnia and Herzegovina	<b>GE</b>	Georgia	<b>MD</b>	Republic of Moldova	<b>TG</b>	Togo
<b>BB</b>	Barbados	<b>GH</b>	Ghana	<b>MG</b>	Madagascar	<b>TJ</b>	Tajikistan
<b>BE</b>	Belgium	<b>GN</b>	Guinea	<b>MK</b>	The former Yugoslav	<b>TM</b>	Turkmenistan
<b>BF</b>	Burkina Faso	<b>GR</b>	Greece		Republic of Macedonia	<b>TR</b>	Turkey
<b>BG</b>	Bulgaria	<b>HU</b>	Hungary	<b>ML</b>	Mali	<b>TT</b>	Trinidad and Tobago
<b>BJ</b>	Benin	<b>IE</b>	Ireland	<b>MN</b>	Mongolia	<b>UA</b>	Ukraine
<b>BR</b>	Brazil	<b>IL</b>	Israel	<b>MR</b>	Mauritania	<b>UG</b>	Uganda
<b>BY</b>	Belarus	<b>IS</b>	Iceland	<b>MW</b>	Malawi	<b>US</b>	United States of America
<b>CA</b>	Canada	<b>IT</b>	Italy	<b>MX</b>	Mexico	<b>UZ</b>	Uzbekistan
<b>CF</b>	Central African Republic	<b>JP</b>	Japan	<b>NE</b>	Niger	<b>VN</b>	Viet Nam
<b>CG</b>	Congo	<b>KE</b>	Kenya	<b>NL</b>	Netherlands	<b>YU</b>	Yugoslavia
<b>CH</b>	Switzerland	<b>KG</b>	Kyrgyzstan	<b>NO</b>	Norway	<b>ZW</b>	Zimbabwe
<b>CI</b>	Côte d'Ivoire	<b>KP</b>	Democratic People's	<b>NZ</b>	New Zealand		
<b>CM</b>	Cameroon		Republic of Korea	<b>PL</b>	Poland		
<b>CN</b>	China	<b>KR</b>	Republic of Korea	<b>PT</b>	Portugal		
<b>CU</b>	Cuba	<b>KZ</b>	Kazakstan	<b>RO</b>	Romania		
<b>CZ</b>	Czech Republic	<b>LC</b>	Saint Lucia	<b>RU</b>	Russian Federation		
<b>DE</b>	Germany	<b>LI</b>	Liechtenstein	<b>SD</b>	Sudan		
<b>DK</b>	Denmark	<b>LK</b>	Sri Lanka	<b>SE</b>	Sweden		
<b>EE</b>	Estonia	<b>LR</b>	Liberia	<b>SG</b>	Singapore		

## INTELLIGENT MODEL FOR 3D COMPUTER GRAPHICS

### Field of the Invention

The present invention generally pertains to three-dimensional (3D) graphic objects, and more specifically, to 3D graphic objects that are provided as models for use in creating 3D scenes and animations.

### Background of the Invention

Collections of two-dimensional (2D) graphic images of various subjects referred to as "clip art" have long been available for users to include in documents and to create graphic scenes on the computer. Typically, such images are catalogued or indexed by subject matter, enabling the user to more readily select the appropriate image from a collection. More recently, 3D objects have also been made available for use in creating 3D scenes and animations. The catalogued data typically defines the shape of a 3D object so that the object can be displayed on the user's screen as a wire frame image that can be rotated and repositioned relative to a viewpoint. In addition, the catalogued data for a 3D object may include texture and color data. Using the texture and/or color data, the wire frame image can then be rendered using Gouraud or other smooth shading rendering techniques to obtain a more natural appearing surface having the specified texture. Catalogued data for 3D objects rarely include information defining characteristics other than the visual appearance of the objects.

When using a conventional 3D object selected from a library catalogue, any behavior associated with the object must be defined by the graphic artist or animator to create a modeled object that responds to input and events that occur during the running of an animation in a realistic (or at least a desired manner). Pixar Animation Studios has developed tools for manipulating a 3D character using controls to define how each part of the character will move in an animation. For example, a control is provided to move an eyebrow or other parts of a 3D character's face, to express an emotion or stylize the character's appearance. However, this approach appears to only simplify the animator's task, since the behavior of the 3D object must still be specified by the animator, although simplified by the tools provided for that purpose.

Certain animation programs, such as the Softimage 3D animation package, also include tools to facilitate associating a behavior with an object to create an animation. However, certain types of behaviors for objects are not readily achieved except by programming the desired behavior in some high

level language such as C or C<sup>++</sup>. In any case, the animator cannot simply select an object and add it to a background in the animation being created without undertaking substantial additional steps to define and implement the desired behavior. Clearly, the time and effort required to write the program code associated with achieving a desired behavior of an object selected from a library of 3D objects adversely impacts the animator's efficiency in creating an animation. It would be much more desirable for an animator to be able to select an object having a full set of associated behavioral parameters from a library of such objects. An object already having a behavior associated with it could then simply be inserted into the animation without requiring that the animator create the desired behavior for the object by writing program code. A library object that includes an associated set of behaviors is referred to herein as an "intelligent model."

A library of intelligent models should include animate objects that have defined attributes and behaviors. Thus, a modeled object with predefined parameters and behaviors could be provided to represent a specific class or type of individual or animal. For example, an anthropomorphic modeled object (male or female) could be associated with certain behaviors that are typical of an athlete, such as the ability to adroitly catch a ball thrown into the vicinity of the object. It would also be desirable to provide a library of packaged behaviors from which a set of behaviors can be selectively associated with a specific type or class of intelligent model. Given a set of associated behaviors, the animator should also be provided with the means to easily edit, extend, or modify parameters affecting the behavior of the intelligent model. Currently, the available libraries of modeled 3D objects do not provide these capabilities or functions.

### **Summary of the Invention**

In accord with the present invention, a modeled object having associated behaviors for use in a graphic image includes shape data defining a shape of the modeled object as it will appear when the shape data are displayed by a computer. Surface data defining an appearance of the surface of the modeled object when displayed by the computer are also provided. In addition, behavioral data are included to define parameters associated with the modeled object that become active in response to a stimulus applied to the modeled object after the modeled object is displayed. The parameters determine a predefined response to the stimulus by at least a portion of the modeled object, so that the modeled object reacts in a desired manner to the stimulus during an animation in which the graphic image is displayed. Thus,

the modeled object can be incorporated into an animation without the user being required to program the object's behavior, since the behavior data for the modeled object already determine how the object will respond to stimuli.

- 5 In one embodiment, the parameters for the modeled object define a behavior that emulates that of a class of selected objects represented by the modeled object. The selected objects may be objects that exist in the real world or imaginative objects. At least a portion of the modeled object reacts to the stimulus in a manner substantially emulating that of an object from the class of selected objects represented by the modeled object. Further, at least a portion of the modeled object may correspond to
- 10 a functional component of the selected object to which the modeled object corresponds, and may respond to the stimulus substantially like the functional component of the real object.

- The model object preferably also includes a plurality of different behavioral data sets from which a specific behavioral data set is selectable by a user for association with the modeled object. The specific
- 15 behavioral data set then comprises different parameters that define different actions of the modeled object in response to a plurality of stimuli. The behavioral data are associated with different portions of the modeled object specified by the shape data, so that a predefined stimulus applied to selected portions of the displayed modeled object triggers a predetermined event.

- 20 An editor, menu option, a or graphic user interface control can be provided for modifying the parameters defined by the behavioral data. In addition, the behavioral data associated with the modeled object are selectively displayable to a user by the computer.

- Further aspects of the present invention are directed to a method for providing a predefined associated
- 25 behavior for a modeled object for use in a graphic image on a computer, and to an article of manufacture for providing a predefined associated behavior for a modeled object. These aspects of the invention are generally consistent with the elements of the modeled object discussed above.

**Brief Description of the Drawing Figures**

The foregoing aspects and many of the attendant advantages of this invention will become more readily appreciated as the same becomes better understood by reference to the following detailed description,

when taken in conjunction with the accompanying drawings, wherein:

FIGURE 1 is an isometric view of a digital computer suitable for implementing the present invention;

FIGURE 2 is a block diagram of some of the more important functional components of the digital computer shown in FIGURE 1;

FIGURE 3 is an exemplary intelligent model representing a television, in accord with the present invention;

FIGURE 4 is a block diagram of the functional components of the intelligent model television of FIGURE 3;

FIGURE 5 is another example of an intelligent model that represents a desk lamp, illustrating how the desk lamp produces light that raises the temperature of a thermometer;

FIGURE 6 is a block diagram illustrating a first embodiment for relating an intelligent model and its run-time library of data to an application that accesses and employs the intelligent model;

FIGURE 7 is a block diagram illustrating a second embodiment for relating an intelligent model and its run-time library to an application that employs the intelligent model; and

FIGURE 8 is a block diagram of yet a third approach for providing an intelligent model to an application.

**Description of the Preferred Embodiment**Computer System Suitable for Implementing the Present Invention

With reference to FIGURE 1, a generally conventional personal computer 30 is illustrated, which is suitable for use in connection with practicing the present invention. Alternatively, a workstation coupled to a network and server may instead be used. Personal computer 30 includes a processor chassis 32 in which are mounted a floppy disk drive 34, a hard drive 36, a motherboard populated with appropriate integrated circuits (not shown), and a power supply (also not shown), as are generally well known to those of ordinary skill in the art. A monitor 38 is included for displaying graphics and text generated by software programs that are run by the personal computer. A mouse 40 (or other pointing device) is connected to a serial port (or to a bus port) on the rear of processor chassis 32, and signals

from mouse 40 are conveyed to the motherboard to control a cursor on the display and to select text, menu options, and graphic components displayed on monitor 38 by software programs executing on the personal computer. In addition, a keyboard 43 is coupled to the motherboard for user entry of text and commands that affect the running of software programs executing on the personal computer.

5

Personal computer 30 also optionally includes a compact disk-read only memory (CD-ROM) drive 47 into which a CD-ROM disk may be inserted so that executable files and data on the disk can be read for transfer into the memory and/or into storage on hard drive 36 of personal computer 30. For use in connection with the present invention, personal computer 30 is preferably coupled to a local area and/or  
10 wide area network and is one of a plurality of such computers on the network.

Although details relating to all of the components mounted on the motherboard or otherwise installed inside processor chassis 32 are not illustrated, FIGURE 2 is a block diagram showing some of the functional components that are included. The motherboard has a data bus 33 to which these functional  
15 components are electrically connected. A display interface 35, comprising a video card, for example, generates signals in response to instructions executed by a central processing unit (CPU) 53 that are transmitted to monitor 38 so that graphics and text are displayed on the monitor. A hard drive and floppy drive interface 37 is coupled to data bus 33 to enable bidirectional flow of data and instructions between data bus 33 and floppy drive 34 or hard drive 36. Software programs executed by CPU 53 are  
20 typically stored on either hard drive 36, or on a floppy disk (not shown) that is inserted into floppy drive 34. The software instructions for implementing the present invention and the data defining a plurality of intelligent models provided in a library will likely be distributed either on floppy disks, on-line via a modem, or on a CD-ROM disk. The data defining the intelligent models comprising a library from which a user or animator can select will typically be stored either locally on the animator's hard  
25 drive 36, or more likely, on a hard drive associated with a server of the network to which the animator's computer or workstation is connected (not shown). When a specific intelligent model is selected, the data defining it will be loaded into the memory of the computer/workstation for use in an animation of scene in accord with machine instructions that define how an application using the intelligent model will apply the behavior of the intelligent model. These machine instructions comprising the application  
30 will also be loaded into memory for execution by CPU 53.

A serial/mouse port 39 (representative of the two serial ports typically provided) is also bidirectionally coupled to data bus 33, enabling signals developed by mouse 40 to be conveyed through the data bus to CPU 53. A CD-ROM interface 59 connects CD-ROM drive 47 to data bus 33. The CD-ROM interface may be a small computer systems interface (SCSI) type interface or other interface appropriate for connection to and operation of CD-ROM drive 47.

A keyboard interface 45 receives signals from keyboard 43, coupling the signals to data bus 33 for transmission to CPU 53. Optionally coupled to data bus 33 is a network interface 50 (which may comprise, for example, an Ethernet™ card for coupling the personal computer or workstation to a local area and/or wide area network).

When a software program is executed by CPU 53, the machine instructions comprising the program that are stored on a floppy disk, a CD-ROM, a server (not shown), or on hard drive 36 are transferred into a memory 51 via data bus 33. Machine instructions comprising the software program are executed by CPU 53, causing it to implement functions determined by the machine instructions. Memory 51 includes both a nonvolatile read only memory (ROM) in which machine instructions used for booting up personal computer 30 are stored, and a random access memory (RAM) in which machine instructions and data are temporarily stored when executing application programs, such as those that use an intelligent model. Examples of such applications programs include graphic animations programs, Web page creation programs, and other environments in which an intelligent model may be employed to implement a set of behaviors or functions.

In connection with the present invention, the term “intelligent model” is synonymous with the concept of a “complete model,” since it encompasses not only the geometry that defines the shape, texture, and general appearance of an object, but also the information defining its construction as a set of rules, the material that comprises it, its animation, and general behavior in response to events. In other words, all of the information required to make use of the model is contained within the data that define the intelligent model. The intelligent model is thus an encapsulation of data and knowledge that define an object and which are made accessible to a user. In addition, the intelligent model can be viewed as an open description format for the model. Furthermore, as will be evident from the examples of pseudo-code associated with defining an intelligent model’s behavior in response to an event that are provided below,

an animator can readily edit the behavior to provide simple modifications. It is thus contemplated that an editor will be provided with one or more intelligent models to enable an animator to open a list of the code instructions associated with an intelligent model and make minor editing changes. Alternatively, the user may select options from a menu, or manipulate a control such as a slider or respond to options in a dialog box. Almost any commonly employed mechanism for enabling user entry and selection of options in either a text or a graphics user interface environment is thus contemplated to enable a user to selectively set one or more parameters for an intelligent model.

A simple example of a first intelligent model is illustrated in FIGURE 3. In this example, a television 70 is illustrated. In appearance, television 70 appears just like prior art 3D models of a television that are not intelligent. Television 70 includes a cabinet or chassis 72, a screen 74, a control panel 76, and a speaker 78. On control panel 76 are disposed a channel selector knob 80, a volume control 82, a contrast control 84, and a brightness control 86 (along with other controls appropriate to a television).

To define the 3D shape and appearance of television 70, as is well known in the prior art, the television will be defined as a collection of polygons (a bounded planar surface). Each polygon is specified by an arrangement of points in 3D space. Plus, chassis 72 would be defined by a series of points generally as follows:

TV = {

Polygon 1 = { (0, 0, 0), (20, 0, 0), (20, 10, 0), (0, 10, 0) };

Polygon 2 = { ... };

...

Polygon N = { ... };

};

The intelligent model's components will be grouped into sub-components or parts. For example, in regard to television 70:

```

TV = {
    Shell = {
        Polygon 1 = { (0, 0, 0), (20, 0, 0), (20, 10, 0), (0, 10, 0) };
        Polygon 2 = { ... };
5        ...
        Polygon N = { ... };
    };
    Screen = {
        ...
10    };
    ...
};

```

In addition to the preceding data that define the shape and general appearance of the intelligent model

15 for television 70, additional information is provided, such as an identification of the surface material and texture for each of the surfaces of the model. For example, screen 74 would have a smooth surface texture and reflective qualities characteristic of glass. Speaker 78 would be defined to have a fabric-like or foam-like texture, while chassis 72 might be defined to have either a wood grain texture or a flat painted texture. This type of information can reside at the model level, as well as at the polygon level.

20 The information included with an intelligent model that sets it apart from simple prior art modeled objects are logical instructions that are executed by a computer to define the behavior of the intelligent model in response to one or more events. These instructions and data are specific to the particular object represented by the intelligent model. For example, for the intelligent model represented by television 70, FIGURE 4 illustrates the functional components of the television that generally define its

25 operation. These functional components, which are generally identified by reference numeral 70', are included in the data defining television 70 and broadly replicate the actual electrical components included in an actual television.

Thus, in connection with the intelligent model for television 70, a block 94 indicates that a line-in signal is

30 provided that is input to a block 96. In block 96, the signal is split into an image component and an audio component. The image component of the signal is input to a block 92 that contains image circuitry used to

produce a picture that will appear on screen 74 consistent with the signal input at block 94. Contrast control 84 and brightness control 86 are coupled to the image circuitry. Thus, the appearance of the image appearing on screen 74 (in an animation in which the intelligent model of television 70 is included) will be controlled by adjustment of contrast control 84 and brightness control 86, just as in an actual television.

- 5 The audio component of the signal from block 96 is applied to audio circuitry in a block 90 of the intelligent model. Since the audio circuitry is coupled to volume control 82, manipulation of the volume control will vary the volume of the sound “produced” by speaker 78. In other words, when the volume control is changed in an animation, the sound associated with a television program in the animation that is heard by someone watching the animation will also be changed. Power for
- 10 “energizing” television 70 is applied from a power source, as indicated in a block 98, and is input to television 70 through a power distribution circuit in a block 100 to the image circuitry in block 92 and to the audio circuitry in block 90.

- Based on the preceding description of the functional blocks associated with the intelligent model
- 15 representing television 70, it will be apparent that the intelligent model is configured to operate and respond to events in a manner similar to an actual television. To define the relationship between the functional components of the intelligent model, a description of each block shown in FIGURE 4 must be provided as part of the data and instructions of the intelligent model.

- 20 A simplified description of the diagram shown in FIGURE 4 looks like:

```
TV( Line_in ) = {  
    Logic = {  
        Audio = get_audio( Line_in );  
        Image = get_image( Line_in );  
5      };  
    Screen = {  
        Geometry = {  
            ...  
        };  
10    apply_texture( polygon 1, Image );  
    };  
    Speaker = {  
        ...  
    };  
15    ...  
};
```

In the context of its use, the intelligent model representing television 70 clearly has greater utility than a prior art simple black box model representing a television. If the animation in which television 70 is included has a character that adjusts volume control 82, the audio level of the television program appearing to emanate from speaker 78 will increase or decrease. Similarly, if an animated character  
5 manipulates channel selector knob 80, the image appearing on screen 74 will change to show a different program image based upon a line-in input applied to block 94 for the channel selected by the animated character. In other words, each of the controls and functionality of intelligent model 70 will correspond to what one would expect for a true television in response to an interaction with an external animated character — all without requiring the user to program the behavior of the intelligent model representing  
10 television 70.

An intelligent model need not be limited to the conventional behaviors associated with an object. Special behaviors can also be provided. For example, television 70 might have a special behavior related to its response to an object that impacts screen 74. In this case, the intelligent model for television 70 may  
15 include an “explode behavior” that will be initiated if the speed of the object and type of object are determined to exceed a level likely to cause screen 74 to implode. If a foam ball impacts screen 74 in a displayed animation, it may simply bounce off without causing any effect. However, if a hard baseball is thrown against the screen, the force and hardness of the object impacting the screen may exceed the level determined to initiate the explode behavior. An animator may choose to use the editor to change the  
20 criteria defining the limiting values for object velocity and object hardness and thus modify the behavior of television 70. The animator may decide to change the behavior from an explosion of the screen to a simple spider web cracking of the screen. Using the provided editor, menu options, or graphic user interface control, the behavior data and instructions that determine a behavior are readily displayed and modified by the animator or user of the intelligent model. For example, the user or animator may change the intelligent  
25 model’s behavior by simply moving a slider to a different position.

FIGURE 5 represents a second example of an intelligent model, a table lamp 120. This intelligent model includes a base 122, an articulated support arm 124, a reflector hood 126, an on/off control 128, and a lamp 130. Intelligent model 120 responds to on/off switch 128 being toggled on by producing  
30 light and heat directed downwardly from lamp 130. Thus, in an animation in which intelligent model 120 is employed, a thermometer 132 (also an intelligent model) that is placed under lamp 130

will respond to the "heat" produced by lamp 130 by increasing the height of its temperature indicating fluid column 134.

When an intelligent model is loaded into an environment such as an animation, the model has to be registered in that environment. The step of registering the model is required so that the environment becomes aware of the model, its properties, and behavior. Because the model may react to events, such as the adjustment of the brightness of the image in the above example, the environment has to notify the model when any event occurs. These events may arise due to actions of other intelligent models (or other conventional modeled objects) in an animation or other application.

10 The instructions for registering a model will generally look like:

Load the Intelligent Model description;

IF the Intelligent Model reacts to an event;

Add Intelligent Model to the list of Intelligent Models subject to that event;

ENDIF;

15 IF the Intelligent Model has time-based behavior:

Add Intelligent Model to the time table;

ENDIF

20 In certain applications of an intelligent model, the user may interact with it by producing an event using the keyboard, the mouse, or other input device. For example, if the user clicks a mouse button while the cursor or pointer associated with the mouse is over a particular portion of the intelligent model being displayed on monitor 38 (FIGURE 1), the intelligent model will be notified of the "mouse click" event and will react to it - but only if an event of that type is defined for that particular intelligent model.

25 Other types of behavior exhibited by an intelligent model may be unrelated to any event. For example, a particular intelligent model may have a radiance level that pulses at a periodic interval at all times while the intelligent model is displayed.

30 In regard to the environment in which the intelligent model is used, the list of intelligent models subject to an event would be referenced each time that an event occurs. Upon the triggering of an event, the list of intelligent models would be scanned, and each intelligent model would be notified of the event. However, it is also contemplated that in certain cases, an intelligent model may be provided with a

behavior enabling it to “consume the event,” thereby keeping the event for itself. In this case, other intelligent models would not be notified of the event. Also, the environment may control the notification of intelligent models based on the type of event that has occurred. Thus, a mouse click event may only be sent to a particular intelligent model that is directly under the mouse pointer, but not to any other intelligent model in the environment. To facilitate this type of event filtering, the configuration will be provided that is appropriate to the particular situation and environment. An example of the pseudo-code instructions for enabling a model to consume an event follows.

```
FOR all Intelligent Models in the list,  
    Pass the event to the Intelligent Model;  
10    IF the Intelligent Model has consumed the event AND  
        the event can be consumed,  
            Clear the event  
            Break out of the loop;  
    ENDIF  
15 NEXT Intelligent Model
```

For certain intelligent models, a dependency on time may be appropriate. In this case, a time table will be provided that contains all the intelligent models in the environment having a dependency on time. The environment will monitor the time and notify the appropriate intelligent models when and if their associated time related events occur. Each intelligent model will then receive a time event and process it accordingly. Since several intelligent models may have to be notified at the same time, a new processing thread must be created for each intelligent model. The intelligent model can then execute at its own pace without blocking any other intelligent model from executing an appropriate response to the time event.

25 The time table can be organized in several different ways, including relative time, absolute time, etc. It will be up to the application in which the intelligent model is used (or the user) to determine how best to organize the time table. In addition, the time table can be sorted in regard to an expiration time for each event to improve its efficiency. The following pseudo-code illustrates a response to the time event in a time table (assumes that no sorting operation has been applied).

```
30 For all entries in the table,  
    IF the time has just expired
```

```
        Send "Time" event to the model;
        IF one-time event,
            Delete the event from the table
        ELSE
5           Re-queue the time event in the table
        ENDIF
    ENDIF
NEXT entry
```

10 It should also be noted that an intelligent model may not have a time-based event as part of its *general* description. Instead, the time event may be dependent upon a particular context. When the intelligent model enters a context having a time dependency, the intelligent model can then be enabled to enter an entry into the time table so that it will respond to the occurrence of that time by implementing a predetermined behavior.

15 To define the logical behavior for an intelligent model, an augmented transition network (ATN) is preferably used. The ATN is basically a state machine that is applied for specifying the logic implemented by an intelligent model, since it describes the various states in which the intelligent model can be. A transition from one state to another is based on a response by the intelligent model to events in the environment of the intelligent model. Upon the receipt of an event, the intelligent model  
20 examines its current state and determines if this event is known and should be responded to. If a response is required for the current event, the intelligent model executes one or more actions that associated with the event, thereby executing a state transition. This transition may move the intelligent model into a new state, or the intelligent model may stay in the same state, dependent upon the user's determination of how the intelligent model should respond. If an event is not defined in the logic that  
25 stipulates the behavior of an intelligent model, it will simply be rejected or ignored. The following pseudo-code illustrates this aspect of the intelligent model.

```
        FOR all of the events associated with the current state,
            IF the current event is the same as the received event
                Execute actions
30             Set new current state
                Break out of the loop
```

ENDIF

NEXT event

Associated with each event to which an intelligent model may respond is a set of actions. An action can be a single statement in the logic definition, like a set of the attributes of the model, or it may be more complex, such as a script that determines a specific behavior in response to an event. The person designing an intelligent model (or the animator) can determine the complexity of the actions of the intelligent model in response to an event, depending upon the result desired. In order to accomplish this result, an animator must be able to address an intelligent model in a series of statements or commands.

Using such commands, the intelligent model and its components must all be addressable and modifiable. The line editor provides the animator or user a simple context for modifying the statements or commands that define the behavior of an intelligent model.

Typically, a first intelligent model will not set an attribute for a second intelligent model that is outside the scope of its own behavior, as defined in the logic data associated with the first intelligent model. A basic premise that applies to intelligent models is that an intelligent model cannot assume the existence of an outside or external environment. The only way that an intelligent model can communicate outside its own scope is to send a message to the environment and to receive events from the environment.

Using that approach, the environment can pass a message received from an intelligent model to other intelligent models that are contained in the environment.

One type of an action associated with an event may be a predefined behavior that is described at the intelligent model level instead of being embedded in the ATN. These behaviors can be viewed as similar to a function call. For example, the designer of an intelligent model might provide an instruction such as WALK (5), meaning that the intelligent model should execute a walk behavior over a distance of five units in response to an event, such as a door opening. To define the walk behavior, a script will be provided that is written in a high-level language such as Visual Basic, JAVA, or C++. However, the animator or user of the intelligent model need never access this script, unless changes are desired, e.g., changing the script to execute the command WALK (4).

When an intelligent model is used, the environment in which it is placed will be responsible for managing a response to the input of an animator, for displaying each of the intelligent models, for dispatching the events upon their generation, and for generating events that are based upon some condition, such as user input, time, etc. Thus, it is important that the environment in which intelligent models are used be enabled to interact with the intelligent models to achieve the desired behavior that is associated with them.

As noted above, the behavior associated with an intelligent model can correspond to that expected from a corresponding real world object, or can be very different. For a given class of intelligent models, different sets of behaviors may be included in the library of intelligent models for selection and association with a selected intelligent model from that class. For example, if the class of intelligent models is "boats," one set of behaviors associated with the boat would provide for the boat to be fast and responsive, while an alternative set of behaviors would cause the boat to appear broken down, enable it to "leak," and to sink. To change the behavior associated with an intelligent model, the user or animator would simply select the intelligent model and load a new set of behaviors from the library, perhaps using a menu item in the editor that is provided. Once selected and associated with the intelligent model, the behaviors would then determine how the intelligent model responds to events and how it is otherwise characterized in its environment.

Behaviors can also be associated with human and non-human characters used as intelligent models. For example, the behaviors that might be associated with a human character represented by an intelligent model could include laughing, jumping, running, etc. Furthermore, characters can be characterized as intelligent models of a specific type and a corresponding set of behaviors selected and associated with that type. For example, a human character (male or female) intelligent model can be associated with a specific set of behaviors appropriate to an athlete. Thus, when the human intelligent object having an associated set of sport behaviors is placed in an environment in which another object, such as a ball, is thrown near the human intelligent model, in response to the ball toss event, the intelligent model can respond by raising a hand to catch the ball. Other sets of behaviors associated with a human character might be those expected of a police officer, or those expected of a soldier. By selecting a different set of behaviors for an intelligent model, the reactions of that intelligent model to events in the environment can readily be modified to meet almost any requirement.

To enable an appropriate response to an event in accordance with the program logic provided for an intelligent model, a run time library 154 can be provided. In one embodiment, run time library 154 is accessed by an application 152 in which an intelligent model 150 is employed. The run time library  
5 includes execution engines that implement specific tasks related to an aspect of the intelligent model. For example, a graphic library 156 will respond to an event by producing a predefined graphical change when indicated by the intelligent model. An image library 160 is an execution engine that knows how to change the image of the intelligent model, e.g., to do an image inversion that will affect the texture of the intelligent model, in response to a request from the intelligent model to do so. Similarly, an audio  
10 library 158 is an execution engine that can play audio and apply effects on an audio stream related to the intelligent model. There is no data contained in these libraries.

The overall functioning of an intelligent model is controlled by a behavior engine 162 included in the run time library. The approach shown in FIGURE 6 is ideal for applications in which the amount of  
15 data defining the intelligent model is required to be relatively small, for example, facilitating transmission of the data over a network through a relatively slow communication path, e.g., via a conventional modem.

If the size of the data required for defining the intelligent model is not a constraint, FIGURE 7 shows an alternative arrangement in which the run time library is maintained at the location where intelligent  
20 model 150' is implemented. When the intelligent model is loaded into an application 152', the run time library associated with the intelligent model is available to it, just as in connection with the approach shown in FIGURE 6. The intelligent model is then responsible for connecting to the run time library and executing its functions correctly. Also, the application must inform the intelligent model of any event occurring or of any object in the vicinity of the intelligent model with which interaction is  
25 required, since these events may trigger a change of state within the intelligent model corresponding to its predefined behavior.

A further embodiment is shown in FIGURE 8. In this embodiment, an intelligent model 194 contains code to be executed 196 on the target platform. For this case, the code to be executed includes only a  
30 subset of the run time libraries, i.e., only the code for the execution engine(s) required for that particular intelligent model to implement its predefined behaviors.

Although the present invention has been described in connection with the preferred form of practicing it, those of ordinary skill in the art will understand that many modifications can be made thereto within the scope of the claims that follow. Accordingly, it is not intended that the scope of the invention in any  
5 way be limited by the above description, but instead be determined entirely by reference to the claims that follow.

The invention in which an exclusive right is claimed is defined by the following:

1. A modeled object having associated behaviors, for use in a graphic image, comprising:
  - (a) shape data defining a shape of the modeled object when the shape data are displayed by a computer;
  - (b) surface data defining an appearance of the surface of the modeled object when displayed by the computer, said surface data including at least one of color and texture; and
  - (c) behavioral data defining parameters associated with the modeled object that become active in response to a stimulus applied to the modeled object after the modeled object is displayed within the graphic image by the computer, said parameters determining a predefined response to the stimulus by at least a portion of the modeled object, so that the modeled object reacts in a desired manner to the stimulus during an animation in which the graphic image is displayed.
2. The modeled object of Claim 1, wherein the parameters for the modeled object define a behavior that emulates a behavior of a class of selected objects represented by the modeled object, at least said portion of the modeled object reacting to the stimulus in a manner substantially emulating that of an object from the class of selected objects represented by the modeled object.
3. The modeled object of Claim 2, wherein at least said one portion of the modeled object corresponds to a functional component of the object to which the modeled object corresponds, and wherein at least said one portion responds to the stimulus substantially like the functional component of said object.
4. The modeled object of Claim 1, further comprising a plurality of different behavioral data sets from which a specific behavioral data set is selectable by a user for association with the modeled object, said specific behavioral data set comprising different parameters that define different actions of the modeled object in response to a plurality of stimuli.
5. The modeled object of Claim 1, further comprising an editor for modifying the parameters defined by the behavioral data.

6. The modeled object of Claim 1, wherein the behavioral data are associated with different portions of the modeled object specified by the shape data, so that a predefined stimulus applied to selected portions of the displayed modeled object triggers a predetermined event.
7. The modeled object of Claim 1, wherein the behavioral data associated with the modeled object are selectively displayable to a user by the computer.
8. A method for providing a predefined associated behavior for a modeled object for use in a graphic image on a computer, comprising the steps of:
  - (a) defining a shape of the modeled object for display by the computer using shape data;
  - (b) defining an appearance of a surface of the modeled object using surface data, when the modeled object is displayed by the computer; and
  - (c) predefining an associated behavior for the modeled object using behavioral data that define parameters associated with the modeled object, said parameters determining a response of at least a portion of the modeled object to a predefined stimulus, so that when the predefined stimulus is applied to at least said portion of the modeled object, the modeled object responds with the associated behavior.
9. The method of Claim 8, wherein the parameters for the modeled object define a behavior that emulates a behavior of a class of selected objects represented by the modeled object, at least said portion of the modeled object reacting to the stimulus in a manner substantially emulating that of an object from the class of selected objects represented by the modeled object.
10. The method of Claim 9, wherein at least said one portion of the modeled object corresponds to a component of the object to which the modeled object corresponds, and wherein at least said one portion responds to the stimulus substantially like the component of said object.
11. The method of Claim 8, further comprising the step of selecting a specific behavioral data set from a plurality of different behavioral data sets for association with the modeled object, said specific behavioral data set comprising different parameters that define different actions of the modeled object in response to a plurality of stimuli.
12. The method of Claim 8, further comprising the step of modifying the parameters defined by the behavioral data to customize the response of the modeled object to stimuli.

13. The method of Claim 8, wherein the behavioral data are associated with different portions of the modeled object specified by the shape data, further comprising the step of applying a predefined stimulus to selected portions of the modeled object to trigger a predetermined event.
14. The method of Claim 8, further comprising the step of selectively displaying the behavioral data associated with the modeled object to a user.
15. An article of manufacture for providing a predefined associated behavior for a modeled object for use in a graphic image on a computer, comprising:
- (a) a memory media adapted for use with a digital computer; and
  - (b) a plurality of machine instructions and data stored on the memory media, said machine instructions and data defining a plurality of functions that are implemented when the machine instructions are executed by the digital computer, said data and functions including:
    - (i) shape data that define a shape of a modeled object for display by the digital computer;
    - (ii) surface data defining an appearance of a surface of the modeled object when the modeled object is displayed by the digital computer; and
    - (iii) behavioral data that define parameters associated with the modeled object, said parameters determining a response of at least a portion of the modeled object to a predefined stimulus, so that when the predefined stimulus is applied to at least said portion of the modeled object, the modeled object responds with the associated behavior.
16. The article of manufacture of Claim 15, wherein the parameters for the modeled object define a behavior that emulates a behavior of a class of selected objects represented by the modeled object, at least said portion of the modeled object reacting to the stimulus in a manner substantially emulating that of an object from the class of selected objects represented by the modeled object.
17. The article of manufacture of Claim 16, wherein at least said one portion of the modeled object corresponds to a component of the object to which the modeled object corresponds, and wherein at least said one portion responds to the stimulus substantially like the component of said object.
18. The article of manufacture of Claim 15, wherein the data and functions further include a plurality of different behavioral data sets from which a specific behavioral data set is selectable by a user

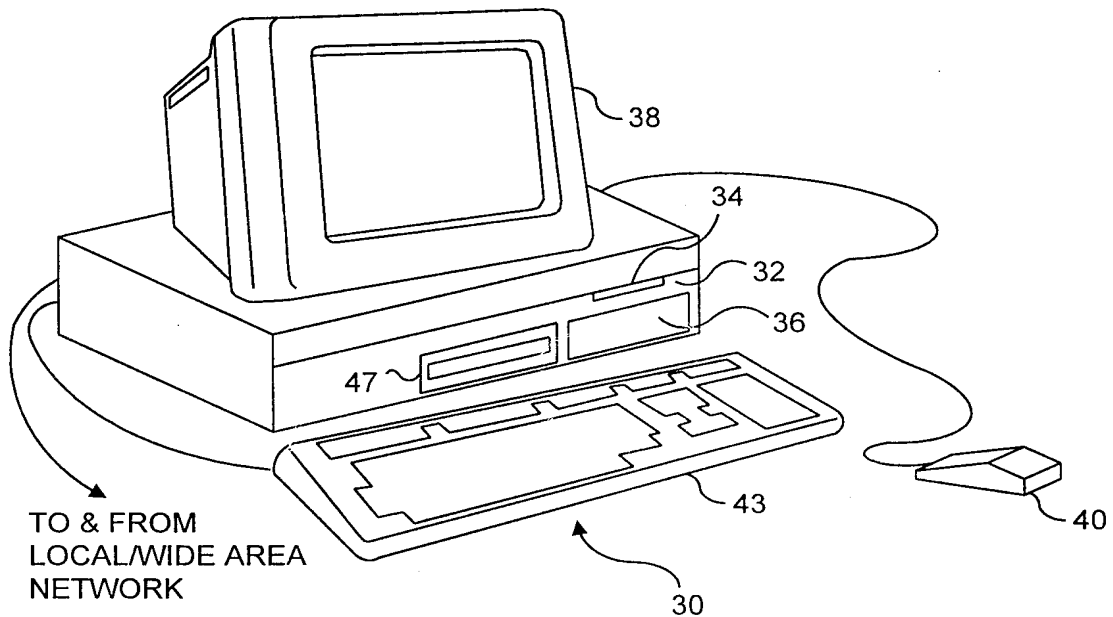
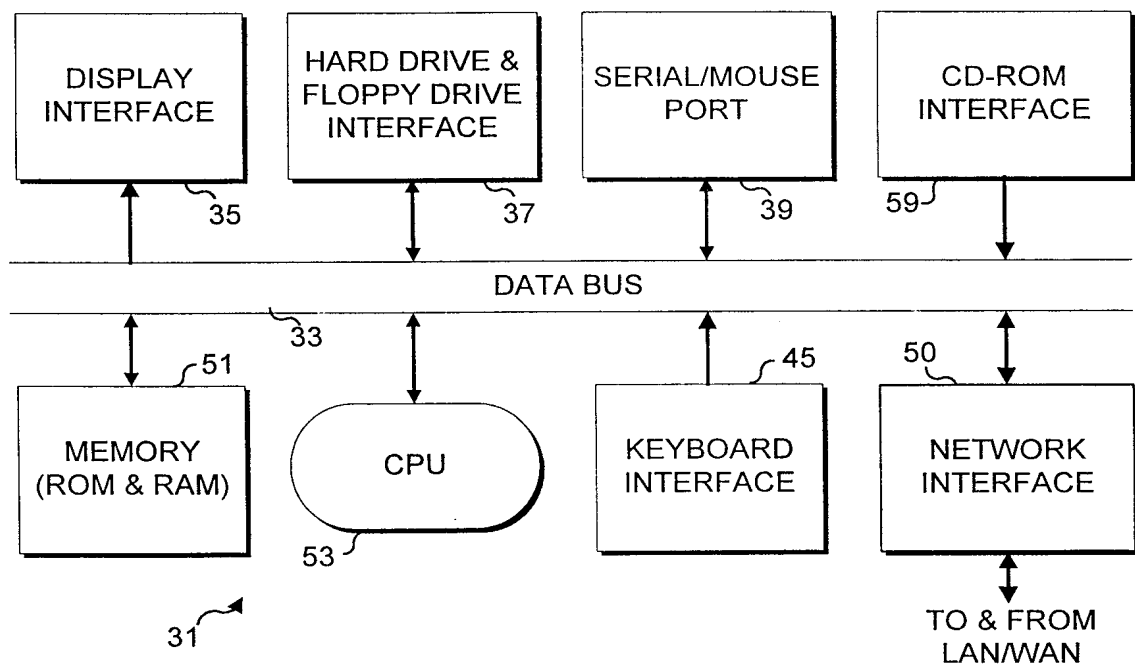
for association with the modeled object, said specific behavioral data set comprising different parameters that define different actions of the modeled object in response to a plurality of stimuli.

19. The article of manufacture of Claim 15, wherein the data and functions further include means for modifying the parameters defined by the behavioral data.

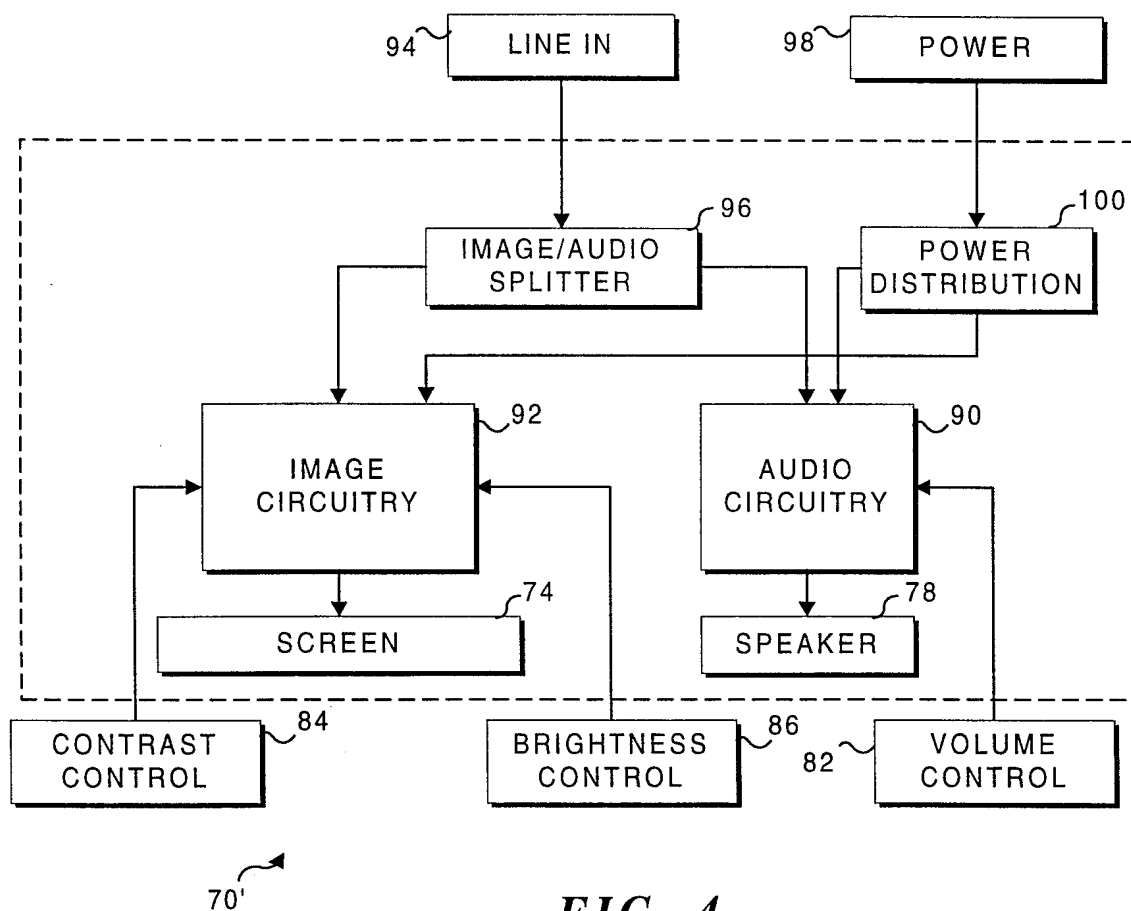
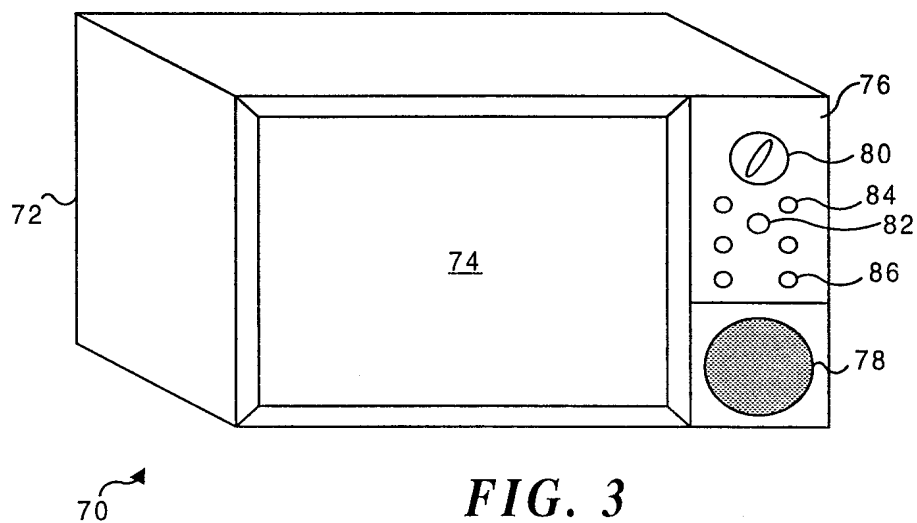
20. The article of manufacture of Claim 15, wherein the behavioral data are associated with different portions of the modeled object specified by the shape data, so that a predefined stimulus applied to selected portions of the displayed modeled object triggers a predetermined event.

21. The article of manufacture of Claim 15, wherein the behavioral data associated with the modeled object are selectively displayable to a user.

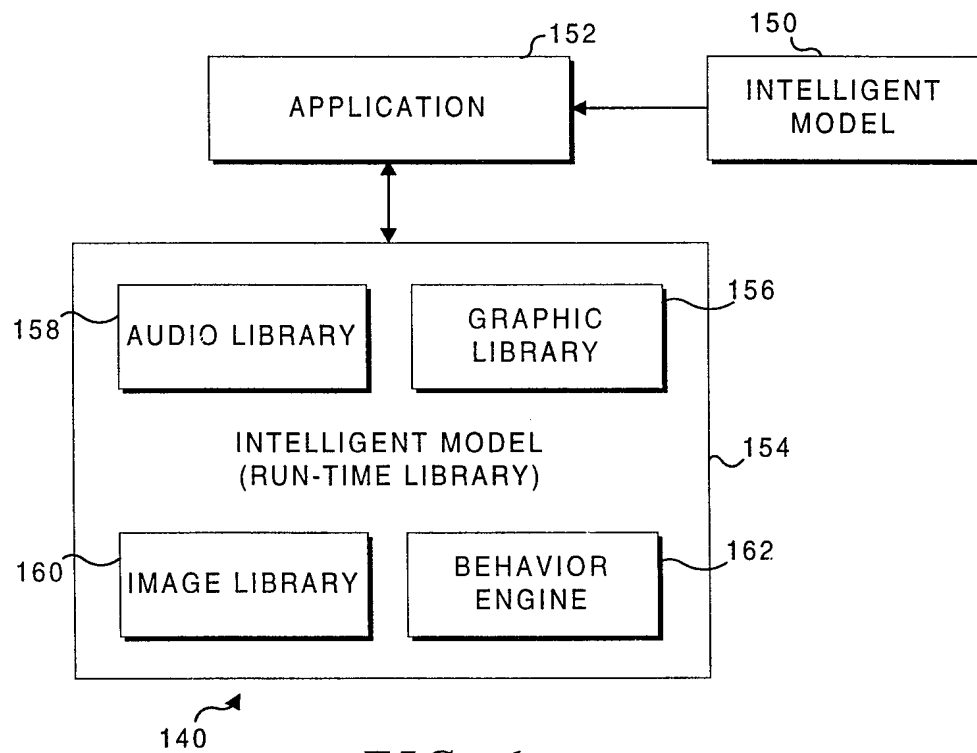
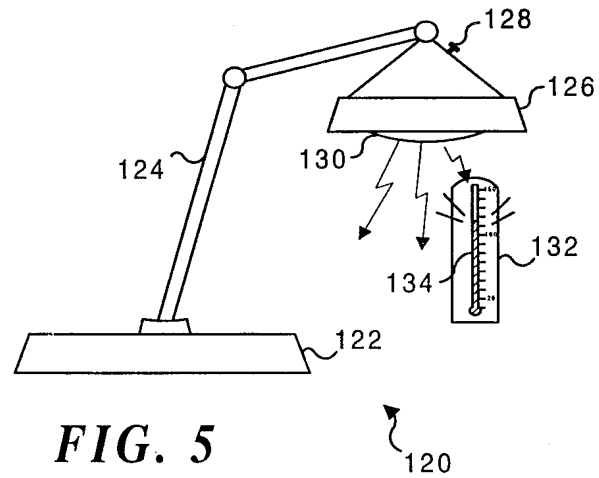
1/4

**FIG. 1****FIG. 2**

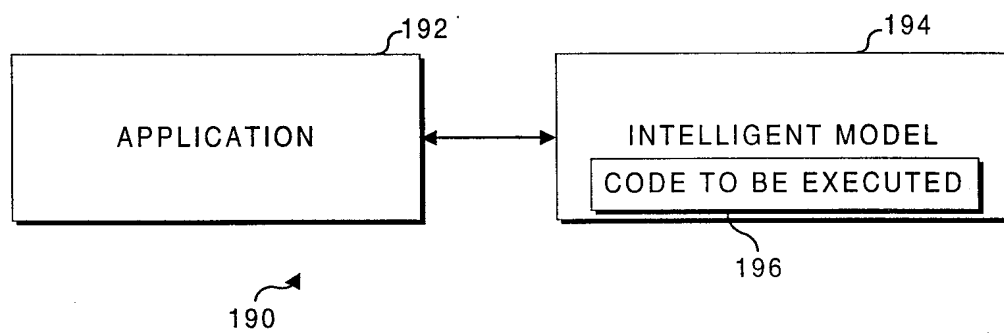
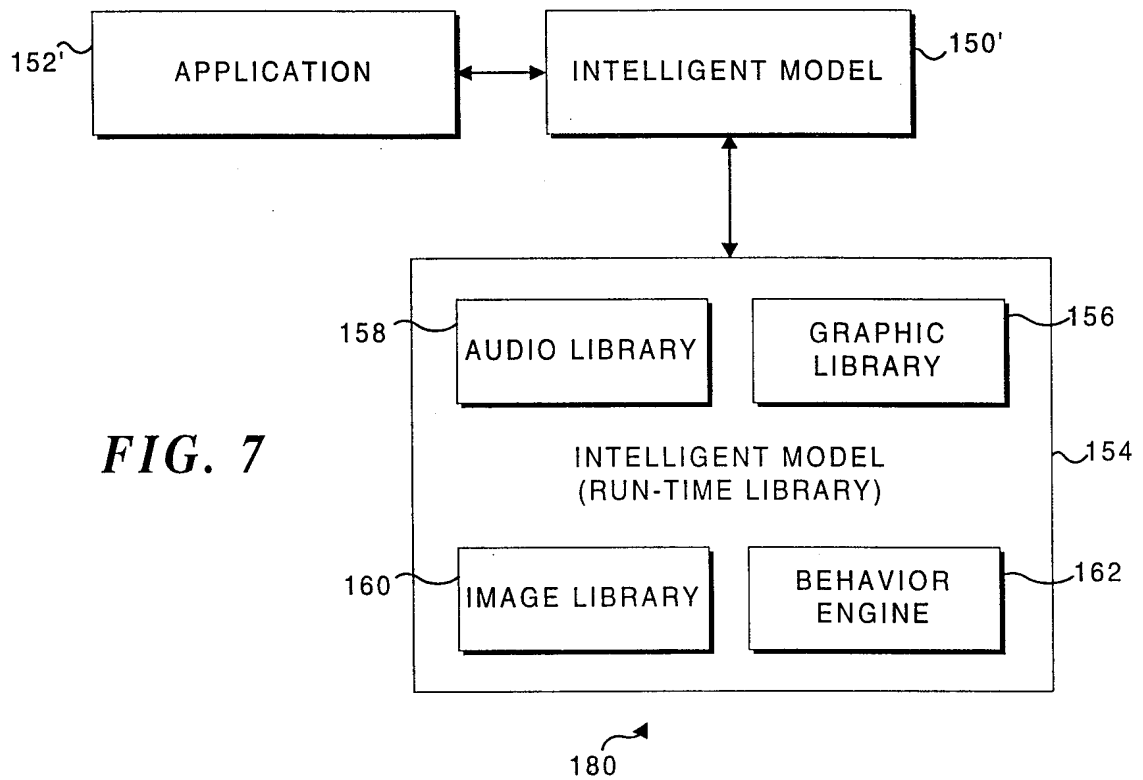
2/4



3/4



4/4



# INTERNATIONAL SEARCH REPORT

International Application No

PCT/CA 98/00696

## A. CLASSIFICATION OF SUBJECT MATTER

IPC 6 G06T17/00

According to International Patent Classification (IPC) or to both national classification and IPC

## B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)

IPC 6 G06T

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Electronic data base consulted during the international search (name of data base and, where practical, search terms used)

## C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	WO 96 23280 A (DALTON NICHOLAS MICHAEL ; PENN ALAN RICHARD (GB); MOTTRAM CHIRON PE) 1 August 1996 see page 10, line 23 - page 13, line 23 see page 20, line 21 - page 21, line 15 ---	1-21
A	KANEKO K ET AL: "Towards dynamics animation on object-oriented animation database system "MOVE" DATABASE SYSTEMS FOR ADVANCED APPLICATIONS '93. PROCEEDINGS OF THE THIRD INTERNATIONAL SYMPOSIUM ON DATABASE SYSTEMS FOR ADVANCED APPLICATIONS, TAEJON, SOUTH KOREA, 6-8 APRIL 1993, pages 3-10, XP002001254 ISBN 981-02-1380-8, 1993, Singapore, World Scientific, Singapore see page 9, paragraph 6; figures 12,13 --- -/-	1-21



Further documents are listed in the continuation of box C.



Patent family members are listed in annex.

### Special categories of cited documents:

- "A" document defining the general state of the art which is not considered to be of particular relevance
- "E" earlier document but published on or after the international filing date
- "L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)
- "O" document referring to an oral disclosure, use, exhibition or other means
- "P" document published prior to the international filing date but later than the priority date claimed

- "T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention
- "X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone
- "Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art.
- "&" document member of the same patent family

Date of the actual completion of the international search

6 November 1998

Date of mailing of the international search report

27/11/1998

Name and mailing address of the ISA

European Patent Office, P.B. 5818 Patentlaan 2  
NL - 2280 HV Rijswijk  
Tel. (+31-70) 340-2040, Tx. 31 651 epo nl,  
Fax: (+31-70) 340-3016

Authorized officer

Perez Molina, E

# INTERNATIONAL SEARCH REPORT

Int. l. Application No  
PCT/CA 98/00696

**C.(Continuation) DOCUMENTS CONSIDERED TO BE RELEVANT**

Category	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A	<p>GAILDRAT V ET AL: "Declarative scenes modeling with dynamic links and decision rules distributed among the objects" GRAPHICS, DESIGN AND VISUALIZATION. IFIP TC5/WG5.2/WG5.10 CSI INTERNATIONAL CONFERENCE ON COMPUTER GRAPHICS - ICCG93, BOMBAY, INDIA, 24-26 FEB. 1993, vol. B-9, pages 165-178, XP002083537 ISSN 0926-5481, IFIP Transactions B (Applications in Technology), 1993, Netherlands see page 172</p> <p style="text-align: center;">---</p>	1-21
A	<p>STRAUSS P S ET AL: "An object-oriented 3D graphics toolkit" SIGGRAPH '92. 19TH ANNUAL ACM CONFERENCE ON COMPUTER GRAPHICS AND INTERACTIVE TECHNIQUES, CHICAGO, IL, USA, 26-31 JULY 1992, vol. 26, no. 2, pages 341-349, XP002083538 ISSN 0097-8930, Computer Graphics, July 1992, USA see page 345, left-hand column</p> <p style="text-align: center;">---</p>	1-21
A	<p>M. GREEN AND S. HALLIDAY: "Geometric modeling and animation system for virtual reality" COMMUNICATIONS OF THE ACM, vol. 39, no. 5, May 1996, page 4652 XP002083539 see page 51, right-hand column, line 26 - page 52, right-hand column, line 8</p> <p style="text-align: center;">-----</p>	1-21

## INTERNATIONAL SEARCH REPORT

### Information on patent family members

In International Application No

PCT/CA 98/00696

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
WO 9623280      A	01-08-1996	NONE	
-----			